ELSEVIER

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 16 (2013) 1043 – 1052

Procedia
Computer Science

Conference on Systems Engineering Research (CSER'13)
Eds.: C.J.J. Paredis, C. Bishop, D. Bodner, Georgia Institute of Technology, Atlanta, GA, March 19-22, 2013.

Facilitating authentic reasoning about complex systems in middle school science education

David A. Joyner, David M. Majerich, Ashok K. Goel

Design & Intelligence Laboratory, School of Interactive Computing, Georgia Institute of Technology, 85 5th Street NW, Atlanta, GA 30332, USA

Abstract

In order to tackle problems in the modern world, individuals must possess a strong ability to reason about and understand complex systems in a practical and useful way. Past research has indicated that experts and novices possess fundamentally different kinds of understanding of complex systems. Therefore, to adequately prepare students to address problems pertaining to complex systems, it is important to help them acquire an authentic expert-like understanding of these systems. We approach this problem from two angles: first, we create an interactive environment in which students may investigate complex systems in a manner similar to that of scientists and engineers; secondly, we embed metacognitive agents in the interactive environments such that the agents provide situated guidance towards expert-like understanding of complex systems. In this paper, we detail the design of these two systems, referred to as MILA and MeTA respectively, and the way in which they help students obtain a more authentic and advanced understanding of complex systems. We also briefly describe the deployments of these tools in a science summer camp for middle school students and preliminary results of students' interaction with them.

Selection and/or peer-review under responsibility of Georgia Institute of Technology

Keywords: Complex systems; education; STEM; intelligent tutoring systems; metacognitive tutoring; problem-based learning; authenticity.

1. Introduction

The modern world is characterized by complex technological and sociotechnical systems. Thus, in order to tackle problems in the modern world, individuals must possess a strong ability to reason about and understand complex systems in a practical and useful way. This has recently led to much research on cognition of systems thinking, and to the development of interactive learning environments to promote understanding and reasoning about complex systems[1][2][3][4].

When we refer to 'complex systems', we mean systems that have multiple types, and a large number, of components and processes, and multiple levels of organization of these components and processes[5][6][7]. By 'systems thinking', we mean expert-like thinking about complex systems characterized by thinking about relationships among components and processes, thinking at multiple levels of abstraction, and thinking about systems as a whole[6][7][8].

In order to support students toward achieving this type of expert-like understanding of the way in which complex systems must be reasoned-over and engineered, we adopt a two-prong approach. We aim to improve students' ability by (a) creating an environment in which they can authentically and personally experience, model, investigate, and

Selection and/or peer-review under responsibility of Georgia Institute of Technology
doi:10.1016/j.procs.2013.01.110

construct an understanding of complex systems, called MILA (Modeling and Inquiry Learning Application), and (b) augmenting that flexible environment with situated agents to guide, critique, and support students' learning, called MeTA (Metacognitive Tutoring Architecture). Although middle school students are not expected to become experts from this limited engagement, the goal is for the structure and style of their reasoning and understanding to begin to more closely resemble practicing scientists. Toward that end, we build on several facets of authentic inquiry, including model-based reasoning[1][2][4][10][11][12]; scientific inquiry[14][15][16][17][18]; and problem-based learning[19]. However, affording students activities to experience these facets is not sufficient on its own to help students acquire an understanding of complex systems; there must also be demonstration, apprenticeship, and guidance. For that, we supplement the learning environment with embedded agents that take on the functional roles of teachers[20][21][22]. While embedded agents that monitor and react to students' interactions are useful devices for teaching new concepts in general, we believe they play a special role in the context of science learning where metacognitive skills such as reasoning about complex systems aid students in demonstrating their understanding of concepts, as well as encouraging reflective thinking and practice.

In this paper, we provide a detailed look at the way in which these two tools – MILA and MeTA – are designed to help students toward achieving expert-like ability to reason about and engineer complex systems. We will also briefly look at the deployments of this interactive tool and its accompanying curriculum over the past summer, followed by a look ahead at future deployments and improvements planned for the system.

2. MILA: Modeling & Inquiry Learning Application

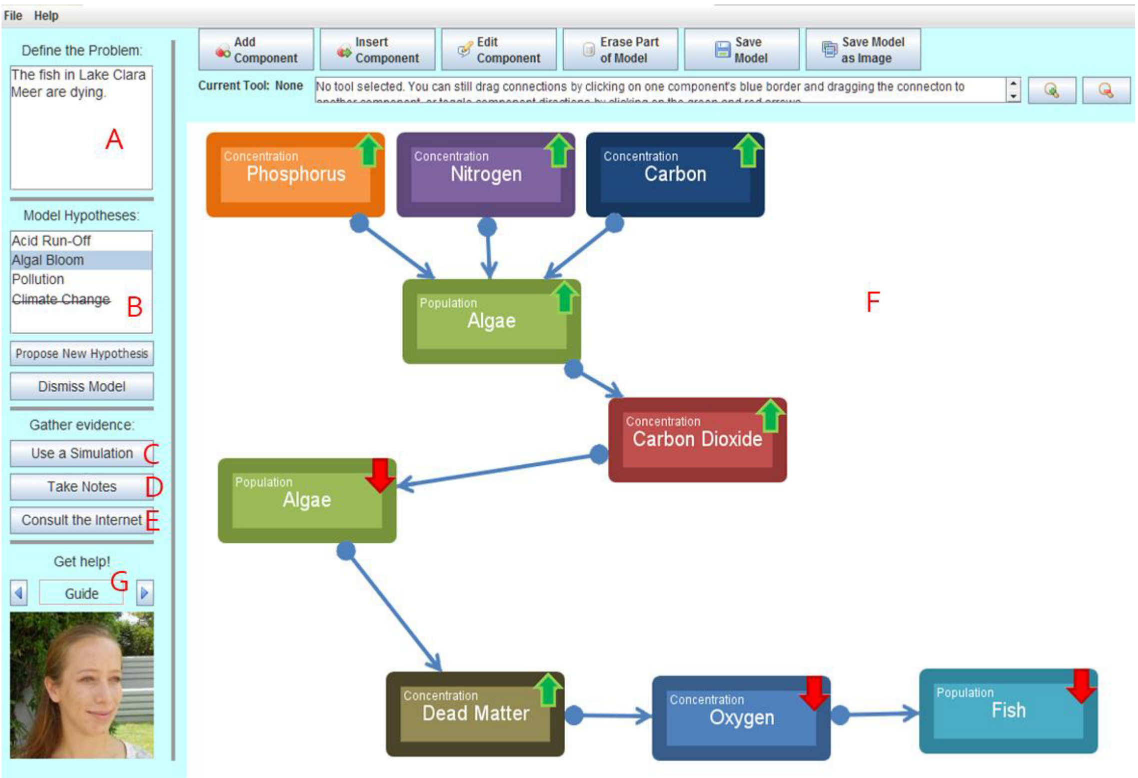

Fig. 1. The MILA software, illustrating the interacting parts.

MILA is presently designed to be situated in the context of a broader curriculum. It supplies the project-based and inquiry-based learning functions that complement classroom lessons, field trips, interviews, and other data-gathering activities. The overall objective of the software layout is to scaffold or facilitate all the tasks associated with authentic inquiry and complex systems reasoning. As such, students follow a five-phase process of: (1) defining their problem; (2) developing initial hypotheses; (3) investigating those hypotheses; (4) redefining, supporting or refuting those hypotheses; and, (5) achieving group consensus for a model that represents the complex system. This workflow is designed specifically to simulate an authentic process of scientific investigation; students pursue an explanatory model for their complex system and its associated problems in the same way that scientists and engineers approach problems in real-world research. MILA's qualitative models are an adaptation of structure-behavior-function models[23][24][25] and MILA itself evolves from the ACT interactive tool[4][26].

2.1. Application Overview

Figure 1 above is a labeled diagram of the core MILA software. The significant portions of the application are:

(A) The problem definition area. In this box, students are asked to define the problem that they are attempting to explain.
(B) The model hypothesis box. Students pose a number of hypotheses for what might be causing the problem that they described in the problem definition box.
(C) The simulation box. Students are provided with a collection of NetLogo simulations to use in their investigations of subsystems within the larger complex system.
(D) The note-taking box. Students are provided with a free-form notepad in which to record observations , and later formalize their notes to become part of their models.
(E) The hypermedia box. Students are provided with restricted internet resources to use in their investigations.
(F) The modeling canvas. Here, students draw a model of how the selected hypothesis might have given rise to the problem that they are explaining. More information on the semantics of these models follows later in the manuscript.
(G) The MeTA tutors. These four tutors "observe" students' interactions and provide responsive advice. More information on the MeTA tutors follows later in the manuscript..

2.2. Application Walkthrough

Upon opening the software, the first task presented to the students is to create a project title corresponding to the problem they have decided to explain. After creating the project, the next task given is to fill in a natural-language description of the system in the top left (A). This problem description then guides all further interactions with the software and investigations of the system. In this way, the problem-driven approach is elevated to an immediate first-class object.

After creating a project and describing the problem, the software opens up into less of a guided approach and more of a flexible environment. While the tasks for students can be presented in any order, the teacher can decide upon the sequence of tasks depending on the objectives and learning goals that are targeted. What follows is a description of how our deployments have utilized our framework.

In our deployments, the first task that students are given is to come up with several initial hypotheses about their problem that they are investigating. In so doing, an emphasis is placed on brainstorming for generating multiple possible explanations of the problem in order to ensure that students have a breadth of potential causes to investigate. More importantly, brainstorming for generating multiple initial hypotheses addresses a learning goal regarding dismissal of hypotheses, detailed later in this walkthrough. For each hypothesis that the students create, they are asked also to come up with a plausible explanation for how that particular hypothesis is related to the problem. For example, in Figure 1 above, four hypotheses are proposed in the box (B): Acid Run-Off, Algal Bloom, Pollution, and Climate Change. The goal for the students is to develop a model for how each hypothesis might be valid under certain conditions.

The bulk of interaction with the environment, then, becomes the authentic process of model construction via scientific inquiry. Students are tasked with using the models they have constructed to guide any additional information they need to gather about the complex system. For example, if students propose that an increase in acid run-off caused the fish to die, they may ask, how did that increase actually lead to the deaths of the fish? In answering this question, the students might propose that there is a causal connection of model (system) components: acid run-off is itself poisonous to the fish, and therefore an increase in its concentration would lead to the deaths of the fish in the lake. Students might also propose a more complex description of model (system) components: the acid run-off might itself cause a change to the ecosystem of the lake, killing off a particular species that is responsible for cleaning other poisonous chemicals from the lake, thus leading to the deaths of the fish. In this way, the software allows the students to start with a simple explanation of the system and scale up to a more complex model.

Models created by students are not simply the products of their own speculation. A major element of the modeling process is for students to provide evidence for how they know that their particular model is accurate. In this way, the need for evidence is itself a scaffold toward achieving a more expert-like understanding of both the scientific inquiry process and complex systems as a whole. Students are taught that there are various different kinds of evidence that can be used, and not all evidence should be considered equal.. Direct observations are useful, but must be validated to be truly convincing. Established scientific theories are typically more convincing, but necessitate that effort be taken to demonstrate that the given theory truly applies to the system in question. Suggestions from experts are better than personal speculation, but experts are equally obligated to defend their proposals with other forms of evidence.

With all these different means of justifying a hypothesis, there of course presents itself the possibility that students will be unable to actually demonstrate and defend a particular hypothesis. This becomes another significant learning goal: the inability to defend a hypothesis is not a negative development. In traditional schooling, students are often presented with the blanket idea that being wrong is bad, but in authentic science, being wrong is inevitable and even beneficial. Thus, students are presented with the notion that there may come a point where they must dismiss one or more of their prior hypotheses as implausible or indefensible. Toward this end, a function is actually provided in the software to dismiss a model, while still preserving its content for future reference.

Armed with these tools, students engage in an iterative process of model-driven data gathering and model construction and refinement. Additional data are used to add more complex mechanistic explanations to their models, or to provide evidence for proposed connections that they have made previously. However, even with the environment in which to pursue this process and develop an authentic notion of complex systems, students could still very easily miss the learning goals. They could develop simplistic models with poor evidence, or fail to use their models to actually inform their continued data-gathering. In order to understand these concepts, students need actual demonstrations and guidance on how to achieve a satisfactory model of a complex system. Toward that end, the MeTA tutors, described in the next major section, are provided.

2.3. Modeling Semantics

The primary intention of the modeling system in MILA is to provide students with a powerful-yet-simple, flexible-yet-constrained representation with which to model and understand complex systems. In order to accomplish this, students are guided toward constructing their understanding in terms of structures and variables in the system and their trends and changes. Thus, each node in the diagram represents a changing variable, as shown in Figure 2.

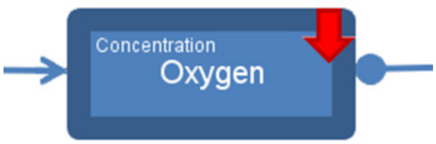

Fig. 2. A node in a MILA model, representing the trend of Oxygen Concentration Decreasing in a system.

Within this node, the large text in the middle represents the physical Component of the system that the node represents. The smaller text in the upper left represents what Variable of that Component is changing. The arrow in the top right represents the nature of that change. So, this node would read, "the Concentration of Oxygen is decreasing."

Nodes are used to represent the causal chains that occur within the system. Therefore, nodes can be linked together by connections. Connections represent causal relationships; the trend described at one end of the arrow (the 'dot' end) causes the trend described at the opposite end (the 'arrow' end). An example is shown in Figure 3.

Fig. 3. A sequence of connected nodes in a MILA model, representing a causal chain of trends.

This chain, then, would read, "The Concentration of Dead Matter goes Up, which causes the Concentration of Oxygen to go Down, which causes the Population of Fish to go Down." In this way, there are very strong natural semantics to the diagrams that students draw; the causal chain and the descriptions of changing trends are inherent and mandatory in the modeling structure. At the same time, students are free to add any physical component and variable they want and draw whatever connections they believe exist, retaining a strong creative and flexible element to the modeling interface.

To tie together this modeling interface, we can look back at the model supplied with the preceding screenshot of MILA. This model can be read as, "Increasing Concentrations of Phosphorus, Nitrogen, and Carbon, lead to Algae Populations increasing, which leads to Carbon Dioxide concentration increasing. However, the increase of Carbon Dioxide Concentration in turn causes Algae Populations to drop, which increases the amount of Dead Matter in the system. As the Dead Matter increases, the amount of Oxygen available decreases, which decreases the Fish Population in the system." Additionally, the modeling framework itself allows for students to create non-linear chains of causation, feedback loops, and other more complex interactions; the models shown in Figures 1 and 3 are examples of the simple models that students construct at first, but the software is constructed to allow more complicated models as students become more comfortable with the software and modeling process.

The final important detail regarding models in MILA is the notion of evidence, seen in Figure 4. A key learning goal in MILA and deployments associated with MILA is the idea that a model is only as good as the defense, argument, and evidence that a student can give for it. Unlike much of traditional schooling, there is no canonical 'correct answer' for students to pursue; instead, students pursue increasingly and iteratively better answers that they can defend with strong evidence. Thus, to support that process, a field is provided in the modeling interface itself to allow students to input the evidence for their models. Clicking on a connection brings up the evidence dialog, where students can enter a natural language description of why they believe a particular connection is true. After entering it, they can collapse the evidence; the connection stays green to indicate that it has been justified with evidence without keeping the evidence box perpetually visible.

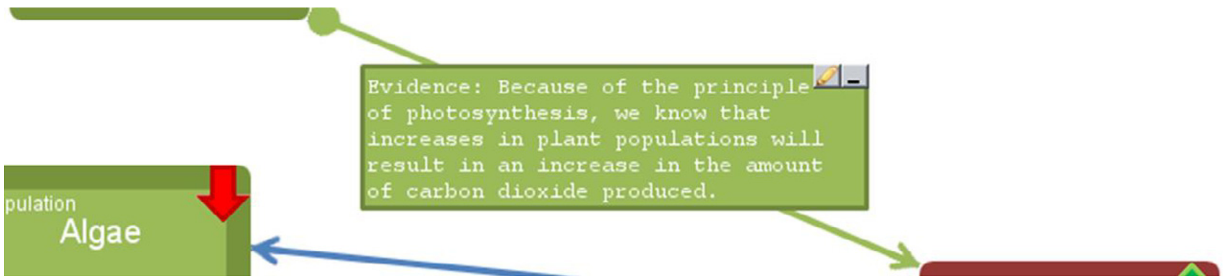

Fig. 4. An evidence box. Evidence is supplied to allow students to give justification for the claims they are making.

3. MeTA: Metacognitive Tutoring Architecture

Broadly, MeTA (again, Metacognitive Tutoring Architecture) is a content-agnostic framework for developing tutoring agents that can observe students' cognitive and metacognitive abilities and react with a variety of feedback options. Metacognitive tutoring presents a number of unique challenges and characteristics that differentiate it from domain-specific tutoring of cognitive skills. Prominently, metacognitive skills are often domain-independent, and thus can be transferred to other domains and environments[27]. For example, a prominent metacognitive tutoring system available presently teaches students the skill of help-seeking and self-assessment, skill that can be transferred and leveraged in a variety of domains[27][28]. Another, deployed in a similar context to our own system, tutors self-regulated learning in the context of biology[29]. MeTA itself is geared toward providing the tools and hooks needed to build tutors for other metacognitive skills. It provides a number of ways to perceive students' actions and behaviors, and the extensibility to add new perceptual abilities for new contexts. It also provides a number of ways of providing feedback to students (and, again, the extensibility to create new such feedback mechanisms), and a flexible infrastructure for pairing perceptions with feedback actions to develop dynamic tutors for new domains.

Within MILA, the MeTA tutors that are included with the software distribution reside in a box in the lower left corner of the software. A brief overview of the inner workings of the system is provided here to facilitate an understanding of how MeTA can help students reason over and engineer complex systems, but for further information, please see the software's dedicated web site.

3.1. Architecture

MeTA is comprised of three primary objects: Percepts, Actions, and Mappings. Percepts are the different ways in which the tutor can perceive the software environment, while Actions dictate how the tutor responds when the particular Percepts are found. Percepts and Actions are combined into Mappings, which then form the bulk of the reasoning that the particular tutor completes; for example, one particular Mapping used in our tutors describes the following: if the student is currently on lesson 7, has created a model, has added a minimum of two nodes to the model, has linked those nodes, and has created a node representing Fish Population, then one of the tutors changes its visible appearance, alerts the student that feedback is available, and alerts the student that their model is sufficient for the learning goals associated with that particular lesson. In discourse, we refer to 'MeTA tutors'. A MeTA tutor is an individual agent implemented in MeTA, of which there can be more than one in a given software context in order to serve multiple functional roles. In our software, we utilize four MeTA tutors: a Guide, a Critic, a Mentor, and an Interviewer. All are implemented in the same architecture, but kept distinct from one another in order to present students with multiple types of feedback and multiple functional roles.

3.1.1. Types of Tutors

With regards to interaction styles, MeTA tutors fall into two broad categories describing the way in which students interact with the tutors. Some tutors are 'Reactive', or 'On-Demand', tutors. These tutors respond to students' help-seeking behaviors. Reactive tutors reside in the tutor box in the software with a visual representation, but only interact with the students when prompted. They will not interrupt the students or check their Mappings on their own.

In contrast, the other available category of tutor is the 'Proactive', or 'Interrupting', tutor. These tutors will actively monitor students' progress and actions, and, when they observe the student doing something that needs intervention, will proactively interrupt the student to provide their feedback or ask their question. Proactive tutors operate by polling their Mappings every 5 seconds; when a Mapping is observed to have true Percepts, the Proactive tutor immediately responds with the Actions associated with that Mapping. Whereas Reactive tutors are present to facilitate help-seeking, Active tutors facilitate just-in-time error correction, and are specifically geared toward correcting misconceptions or providing guidance when students may not otherwise recognize the need for such intervention.

3.2. Content

MeTA as an architecture is designed to be as flexible as possible; it is not designed to have any direct connection to MILA or the MILA project, but rather it is designed to be extensible and applicable to any Java desktop application. The meat of the architecture comes in the way in which it is used in a given curriculum, classroom context, or software environment. MeTA tutors are designed to facilitate multiple functional roles and actions in the classroom context. Choosing which roles to be specifically developed is a task for the designer of the curriculum or classroom context.

With regards to our deployments revolving around increasing students' understanding of complex systems, MeTA tutors were geared toward a number of notable functional roles. Several of these functional roles are outlined below. Note that these functional roles are the product of both advanced planning and retrospective data analysis.

Some of these were included in our tutors as deployed in the intervention described later in this paper, while others were observed to be needed at some point during that intervention. Note that these also represent only a small subset of the roles played by tutors throughout the intervention; several other roles, less related to systems engineering research, were developed or observed as well.

- Demonstrating Concepts. In many instances, students had trouble articulating or specifically demonstrating a certain underlying notion in an external way. For example, students sometimes had trouble with the temporal nature of MILA models. A functional role of the tutors, then, becomes to demonstrate, using a combination of natural language instruction and software demonstration, the way in which a certain complex series of temporal changes is represented within the framework. This does not amount to simply a software tutorial; this, in turn, feeds into the tutors' role in guiding students toward a more complete understanding of the complicated dynamics at work in complex systems.
- Facilitating Reflection. A major element of metacognitive learning, in complex systems engineering and any other field, is the notion of reflection. In order to develop a personal and complete understanding of any skill, it is imperative that the learner reflect on their own understanding and learning process. Oftentimes, however, encouraging such reflection can be difficult, and validating that it has occurred can be even more difficult. As such, another functional role played by the tutors is to actively encourage or mandate personal reflection on one's understanding. This takes the form both of prompting the reflection and in aggregating the results of such reflection, in order to help the teacher or instructor understand students' current challenges and strengths.
- Scaffolding Content Knowledge. A common problem with intelligent tutoring initiatives is that their main responsibility is in intervention and correction; as such, students quickly begin to regard tutors as assessment tools or monitors for mistakes rather than as actual partners in the learning process. In order to combat this, these tutors are specifically designed to play a positive functional role from the students' perspective. Any time the student presents an idea for which the tutor can provide defense or support, the tutor also intervenes. In this way, tutor intervention is often a positive act rather than simply a negative one.
- Validating Progress. Lastly, one of the significant learning goals for students in learning about complex systems is to formulate a personal understanding of the 'success' point; in other words, at what point can we be somewhat satisfied with our model and hypothesis? Toward this end, the tutors also serve the role of validating students' progress. They reflect on students' current models and provide insight into the strengths and weaknesses. This action is not simply a surface-level reflection on the students' models, but rather serves the broader purpose of helping scaffold students toward an understanding of the criteria for a good model of a complex system altogether.

3.3. Our Tutors

During our deployment, we included four tutors created in MeTA in MILA: a Guide, a Critic, a Mentor, and an Interviewer. The Guide and Critic are Passive tutors, while the Mentor and Interviewer are Active tutors. All four tutors were created with the intention of duplicating or aiding a functional role that a teacher typically plays in the classroom. It is important also to note these tutors were developed specifically for our curriculum; they are not meant to apply identically to other units.

The Guide serves to anticipate questions that the student might want to ask. When prompted, it perceives the current lesson that students are on, the recent activities that the students have completed, and the current status of the students' model. Then, in response to those elements, it presents students with a list of questions they might be interested in asking. When students click a question, they are led either to an additional set of questions or to the answer to their question.

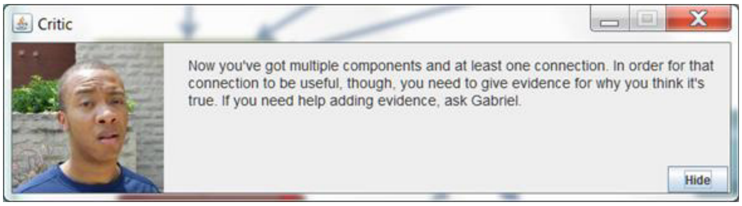

Fig. 5. Craig the Critic, one example of a MeTA tutor.

When clicked, the Critic gives feedback specifically on students' models. It perceives the current structure of the model, the number of nodes and connections, the arrangement, and the presence of evidence. It also perceives some content elements of models, telling student when certain fundamentally needed components are presently absent from their models.

The Mentor serves a dual purpose. First of all, whenever students open the software, the mentor opens to tell students the learning goals for this particular lesson. This is intended to set clear expectations for the content and activity that will be seen during that particular lesson. Secondly, the Mentor monitors for notable developments in the modeling process that need immediate feedback. For example, if a student creates a component that refers to "pH Amount", the Mentor will alert the student that he has feedback and let him or her know, "It's more proper to refer to that as pH level." In this way, the guidance is provided just-in-time when the student has the best chance of absorbing and remembering the correction.

Functionally, the Interviewer pops up and asks the students to type in natural language responses to particular questions that it has during the engagement with the software. This also serves a dual purpose. First of all, this is meant to encourage reflective practice; by specifically asking students to reflect on their models and modeling process and write an answer to a question, the Interviewer reinforces their meta-reasoning about modeling and complex systems. Secondly, and more pragmatically, the Interviewer is also a way in which researchers can gather data about the invisible thought processes in which students engage while modeling.

4. Results

MILA, along with the MeTA tutors, was deployed in the context of a two-week summer camp run at Georgia Tech. The overarching goal of the camp was for students to explain why several thousand fish spontaneously died all at once in a nearby lake a few years ago. As part of this camp, they investigated the lake (both in person and through research using second-hand sources), utilized simulations, conducted experiments, and interacted with MILA and MeTA. All camp activities were driven by the goal of creating a strong explanatory model for the events that transpired in the lake. In actuality, the events that gave rise to the fish die-off (an algal bloom) present an excellent example of a very complex system, comprised of many interacting parts and chains of causality, in action.

4.1. Modeling Results

Using MILA, the six different groups of students (sixteen students, total) created a variety of explanatory models to represent what gave rise to the phenomenon. Most had a strong level of ecological and scientific accuracy; confronted with numerous data sources, nearly every group came to some form of the conclusion that the fish had died from suffocation as an overgrowth of bacteria grew to consume the dead algae. More advanced statistical analysis of these models is ongoing.

4.2. Metacognitive Tutoring Results

In addition to analyzing the models themselves that come out of the summer camp, we are also in the process of analyzing the patterns of interaction that took place between the students and the MeTA tutors. The goal behind this research is to identify a set of design guidelines that can be deployed to ensure the development of tutors that successfully and effectively scaffold authentic and personal understanding of complex systems. As part of this, we ask how interaction with the MeTA tutors changes the conversations and models that students are producing, and what further opportunities there are for MeTA tutors to intervene and provide useful in-context feedback to the students. One way in which we aim to answer these questions is through a detailed analysis of what prompts students to seek help; the timeline below is an example of one such analysis, detailing the different patterns of interaction with the tutors different groups engaged in over time.

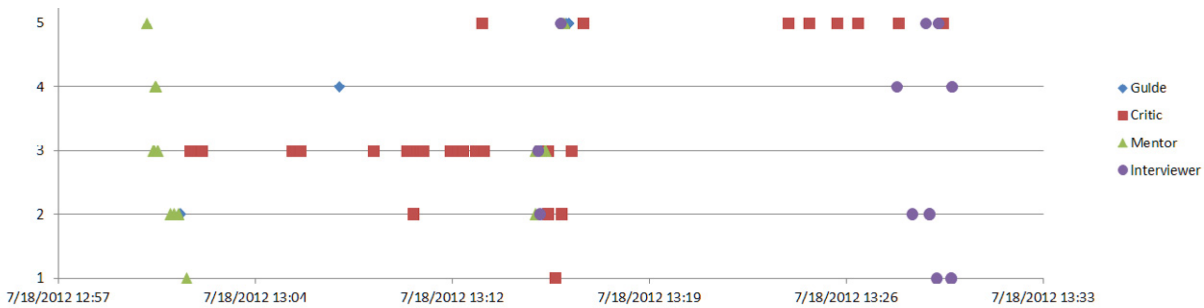

Fig. 6. A timeline view of students' interactions with the four tutors during one particular lesson of the curriculum.

5. Conclusion & Future Work

In order to be prepared to attack the complex problems associated with advanced education and careers, students must be given opportunities to develop expert-like reasoning about complex systems. However, too often in mainstream education, classes underemphasize the ability to think and reason about complex systems in an authentic and practical way. Students instead are often tasked with learning enormous amounts of rote material, but the actual reasoning process that they utilize is not a first-class learning goal. Curricula and assessments are often driven by content knowledge, and while content knowledge is important, the metacognitive skills associated with reasoning about complex systems ought to be first-class learning goals. In addition to better preparing students for the next level of education or career performance, these types of skills are also inherently more motivating for students[14]; learning in the context of real problems with real observable outcomes presents a much more significant motivator towards mastery learning than simple memorization and repetition of content knowledge[19].

With these observations in mind, MILA and MeTA are situated in the context of problem-oriented, model-based science classes in order to help bridge the gap between classroom learning and authentic scientific practice. They aim to supply two of the most significant elements often absent from traditional science classrooms: an authentic and flexible inquiry environment and intelligent, situated guidance and apprenticeship. These two tools attempt to create in a classroom a simulation of how 21st century science is practiced, where scientists learn from each other in the context of the problems they are actively attempting to solve. Combining these tools together, we aim to supply an engaging and authentic learning environment toward the development of better metacognitive skills through increased student engagement, motivation, and interest in science

Moving forward, numerous improvements are planned for both MILA and MeTA in order to enhance and generalize their applicability to a variety of fields. The modeling framework in place in the interactive tool at present captures emergent ecological and chemical phenomena, but is relatively weak when applied to other domains. Efforts are planned to extend the tool with frameworks better suited for a wider variety of subject matter, including biology and chemistry. On the MeTA side, many of the future plans involve not only enhancing the tutor's role within MILA, but also extending it into other environments as well. Within MILA, future tutors in the MeTA architecture will provide better feedback, more intelligent reactions, and more fine-grained percepts and understanding of students' models and progress. Beyond MILA, though, plans are being laid to create MeTA tutors in other contexts. One particular context in which MeTA tutors may be deployed in the near future is toward design teams tackling complicated systems engineering problems. Here, a MeTA tutor may play the role of a team member, reading and reacting to other team members' insights.

Acknowledgments

We are grateful to Kristina Strickland, the science teacher for the science camp in which we deployed MILA and MeTA during the summer of 2012. We are grateful also to Georgia Tech CEISMC, especially Andrew Burch and

Chris Thompson, for their support in organizing the camp, and to Nicolas Papin and Cherish Weiler for their help in running the camp. Finally, we thank Mary Moerlins (Piedmont Park Conservancy), Helen Ellis (Georgia Aquarium), and Drs. Jeannette Yen, Marc Weissburg, and Megan Duffy (Georgia Institute of Technology) for providing summer camp science activity support. We thank Proctor & Gamble for an educational grant that partially supported this work.

References

1. Biswas, G., Leelawong, K., Schwartz, D., & Vye, N. (2005). Learning By Teaching: A New Agent Paradigm For Educational Software. *Applied Artificial Intelligence* 19(3-4): 363-392.
2. Bredeweg, B., Linnebank, F., Bouwer, A., and Liem, J. Garp3 - workbench for qualitative modelling and simulation. Ecological Informatics, 4(5-6):263–281, 2009.
3. Bridewell, W., Sanchez, J., Langley, P., & Billman, D. (2006). An interactive environment for the modeling and discovery of scientific knowledge. Human–Computer Studies, 64, 1099–1114.
4. Vattam, S., Goel, A., Rugaber, S., Hmelo-Silver, C., Jordan, R., Gray, S, & Sinha, S. (2011) Understanding Complex Natural Systems by Articulating Structure-Behavior-Function Models. *Journal of Educational Technology & Society,* 14(1): 66-81.
5. Forrester, J. (1968) *Principles of Systems*, 2nd ed. Pegasus Communications.
6. Simon, H. (1962). The architecture of complexity. In Procs. American Philosophical Society, 106(6), 467-482.
7. Simon, H. (1999). Can there be a science of complex systems? In Y. Bar-Yam (Ed.), *Unifying themes in complex systems* (pp. 3-14). Cambridge, MA: Perseus.
8. Silk, E. & Schunn, C. (2008). Core concepts in engineering as a basis for understanding and improving K-12 engineering education in the United States. Paper presented to the *National Academy of Engineering/National Research Council workshop on K-12 Engineering Education*, Washington, D.C.
9. Stewart, J., Cartier, J., & Passmore, C. (2005). Developing Understanding Through Model-based Inquiry. In M. S. Donovan & J. Bransford (Eds.), *How people learn*. Washington D.C.: National Research Council.
10. Bridewell, W., Sanchez, J., Langley, P., & Billman, D. (2006). An interactive environment for the modeling and discovery of scientific knowledge. Human–Computer Studies, 64, 1099–1114.
11. Clement, J. (2008). Creative Model Construction in Scientists and Students: The Role of Imagery, Analogy, and Mental Simulation. Dordrecht: Springer.
12. Lehrer, R., & Schauble, L. (2006). Cultivating model-based reasoning in science education. In R. Sawyer (Ed.), The Cambridge handbook of the learning sciences. New York: Cambridge University Press.
13. Stewart, J., Cartier, J., & Passmore, C. (2005). Developing Understanding Through Model-based Inquiry. In M. S. Donovan & J. Bransford (Eds.), *How people learn*. Washington D.C.: National Research Council.
14. Shaffer, D. W., & Resnick, M. (1999). "Thick" Authenticity: New Media and Authentic Learning. Journal of Interactive Learning Research, 10(2), 195–215. Retrieved from http://eric.ed.gov/ERICWebPortal/recordDetail?accno=EJ591695
15. Duschl, R. & Grandy, R. (2008) Teaching scientific inquiry: Recommendations from research and implementation. Rotterdam, Netherlands: Sense Publications.
16. Quintana, C., Reiser, B., Davis, E. A., Krajcik, J., Fretz, E., Duncan, R., Kyza, E., Edelson, D., & Solloway, E. (2004). A scaffolding design framework for software to support science inquiry. Journal of the Learning Sciences, 13(3), 337-386.
17. Edelson, D. (1997). Realizing authentic scientific learning through the adaptation of scientific practice. In K. Tobin & B. Fraser (Eds.), *International Handbook of Science Education*. Dordrecht, NL: Kluwer.
18. Nersessian, N. (2008). *Creating Scientific Concepts*. Cambridge, MA: MIT Press.
19. Reiser, B. (2004). Scaffolding Complex Learning: The Mechanisms of Structuring and Problematizing Student Work. Journal of the Learning Sciences, 13(3), 273-304. doi:10.1207/s15327809jls1303_2
20. Schwarz, C., & White, B. (2005). Metamodeling Knowledge: Developing Students' Understanding of Scientific Modeling. Cognition and Instruction, 23(2), 165-205.
21. Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation.
22. VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2005). The Andes physics tutoring system: Lessons learned. International Journal of Artificial Intelligence in Education, 15(3), 147–204.
23. Bhatta, S., & Goel, A. (1997). Learning Generic Mechanisms for Innovative Design Adaptation. *Journal of the Learning Sciences,* 6(4): 367-396.
24. Goel, A., Gomez, A., Grue, N., Murdock, W., Recker, M., & Govindaraj, T. (1996). Towards Design Learning Environments - Exploring How Devices Work. *In Proc. International Conference on Intelligent Tutoring Systems*, Montreal, Canada, June 1996.
25. Goel, A., & Stroulia, E. (1996). Functional Device Models and Model-Based Diagnosis in Adaptive Design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Special Issue on Functional Representation and Reasoning, 10:355-370.
26. Goel, A., Vattam, S., Rugaber, S., & Joyner, D., Hmelo-Silver, C., Jordan, R., Honwad, S., Gray, S., Sinha, S., (2010) Learning Functional and Causal Abstractions of Classroom Aquaria. In *Proc. 32nd Annual Meeting of the Cognitive Science Society*, Portland, Oregon, August 2010.
27. Roll, I., Aleven, V., McLaren, B., & Koedinger, K. (2006). Designing for metacognition—applying cognitive tutor principles to the tutoring of help seeking. Metacognition Learning.
28. Roll, I., Aleven, V., McLaren, B., & Koedinger, K. (2011). Metacognitive Practice Makes Perfect: Improving Students' Self-Assessment Skills with an Intelligent Tutoring System. AIED 2011: 288-295.
29. Azevedo, R., Witherspoon, A., Graesser, A., McNamara, D., Chauncey, A., Siler., E., Cai, Z., Rus, V., & Mintean, M. (2009). MetaTutor: Analyzing Self-Regulated Learning in a Tutoring System for Biology. AIED 2009: 635-637.